TCS
Preliminary Design Review

Software
Top Level Design
Section 4

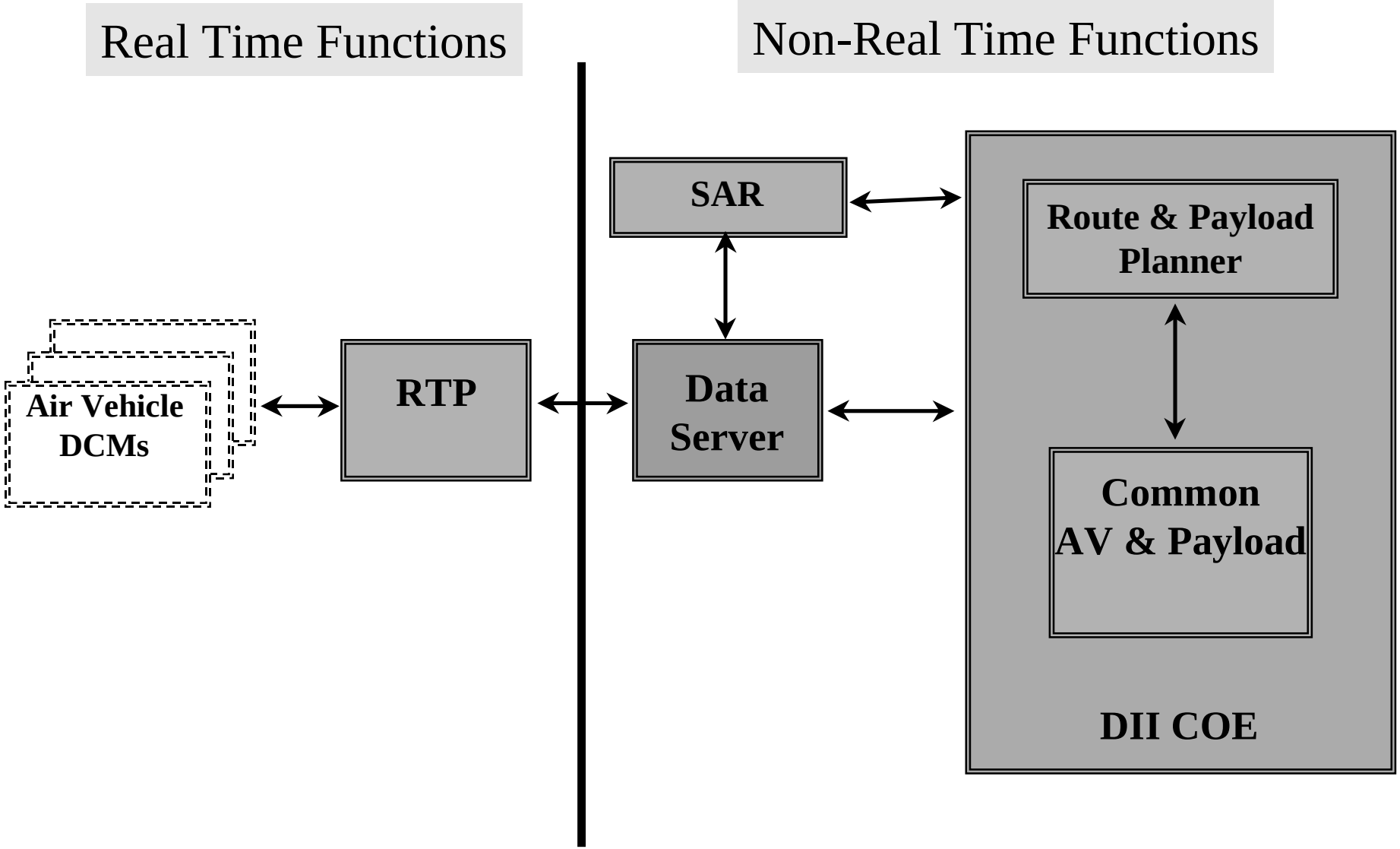
February 1997

NAVAL SURFACE WARFARE CENTER
DAHLGREN DIVISION

Agenda

- PDR Part I Review
- Operational Model
- Target Configuration/Environment
- Top Level Design
  - Section 1: CSCIs Overview
  - Section 2: DCM CSCIs
  - Section 3: RTP CSCI
  - Section 4: DataServer CSCI
  - Section 5: Common AV & Payload CSCI
  - Section 6: DII CSCI
  - Section 7: Route & Payload Planner CSCI

TCS Block 0 CSCI Diagram



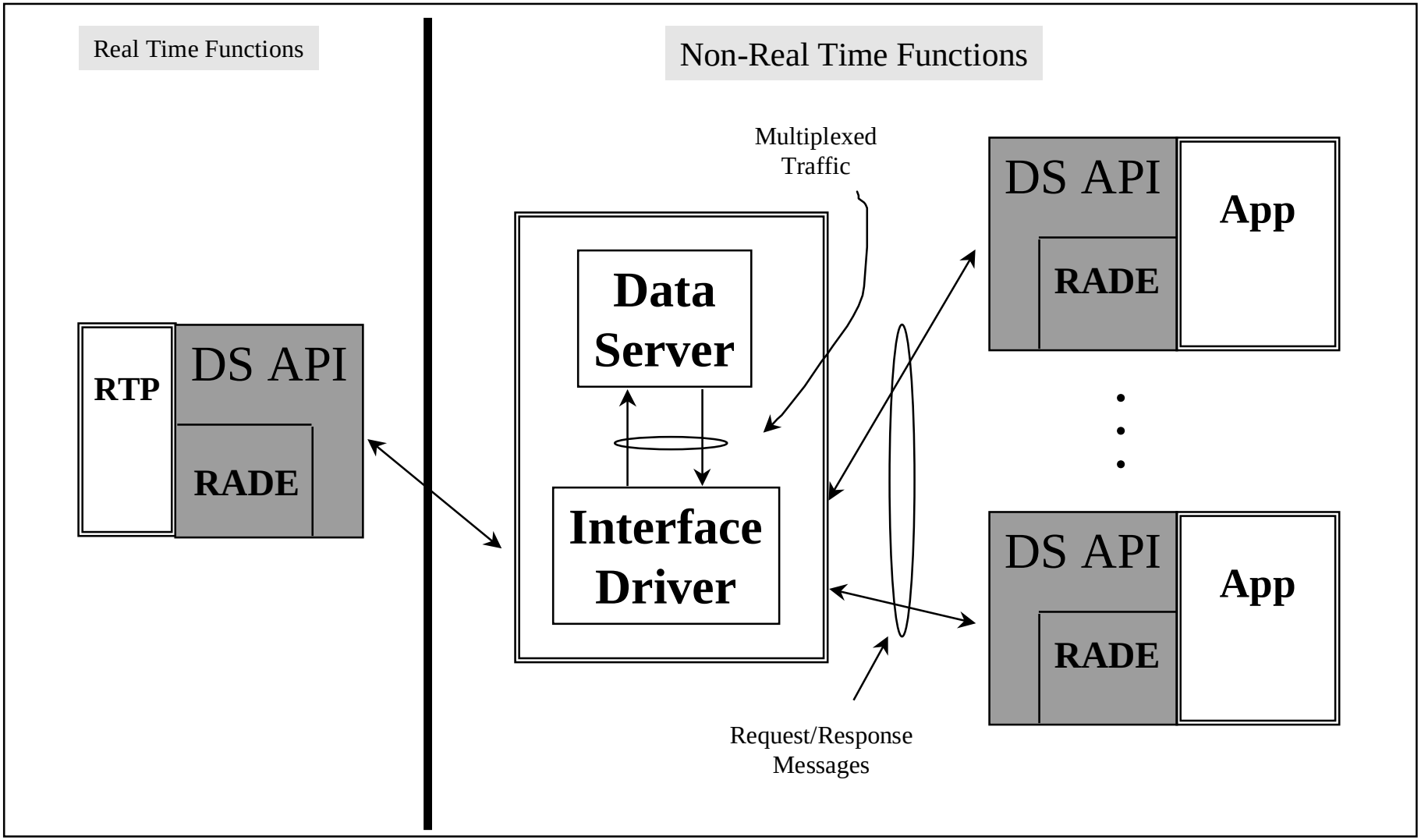
Data Server CSCI

- Description
  - Provides a data sharing capability to programs running on a distributed system
  - Provides a mechanism to transfer data in a machine independent manner
  - Central repository for data that “moves” though the system
  - Client Server based architecture
  - Provide data logging
  - Provides a mechanism to forward data changes to clients automatically
  - Provides persistent storage for clients to recover from an “abnormal situation”

Data Server CSCs

- CSCs
  - Data Server CSC
  - Interface Driver CSC
  - Data Server API CSC

DataServer Block Diagram



DataServer CSC

- Description
  - Manages all client request and makes available the data to other clients
    - Creates buckets as specified by clients
    - Provides automatic data forwarding
    - Supports data protection via write key
  - Provides logging of data as specified by client
  - Supports automatic recovery of data after an abnormal shutdown

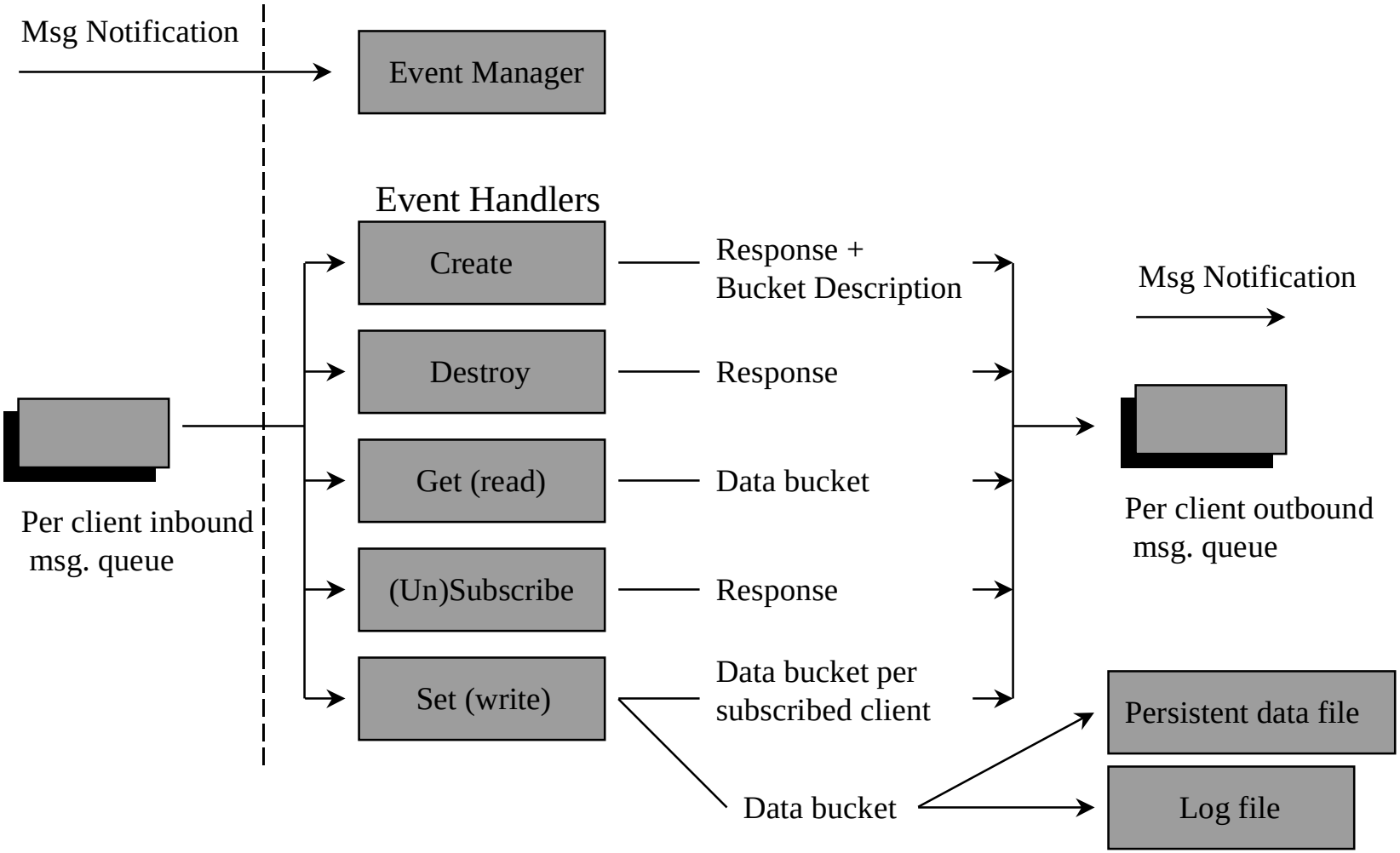
Data Server CSC (cont)

- Interfaces
  - if_driver through pipes and shared memory
  - Disk files
- Inputs
  - Schemas
  - Persistent data file
  - Request messages
  - Command line arguments
- Outputs
  - Response messages
  - Persistent data file
  - Log file

Data Server CSC (cont)

- Startup
  - Initialize data structures
  - Read persistent data file if on exists
  - Open log file
  - Start the if_driver
- Processing
  - Wait for event notification from the if_driver
  - Decode message and route to the appropriate handler
  - Forward response to requsting client
  - Forward data to subscribing client
- Shutdown
  - Close persistent data file, close log file, terminate if_driver execution

Data Server CSC Block Diagram



Interface Driver CSC

- Description
  - I/O Interface for the Data Server CSC
    - Socket based I/O Multiplex/De-multiplex
    - Manages all client connections
  - Provide a mechanism to meet clients “burst data” needs and to ensure connectivity
  - Provides a mechanism to accommodate ‘slow reading clients’
- Interfaces
  - Data Server through pipes and shared memory
  - Client processes

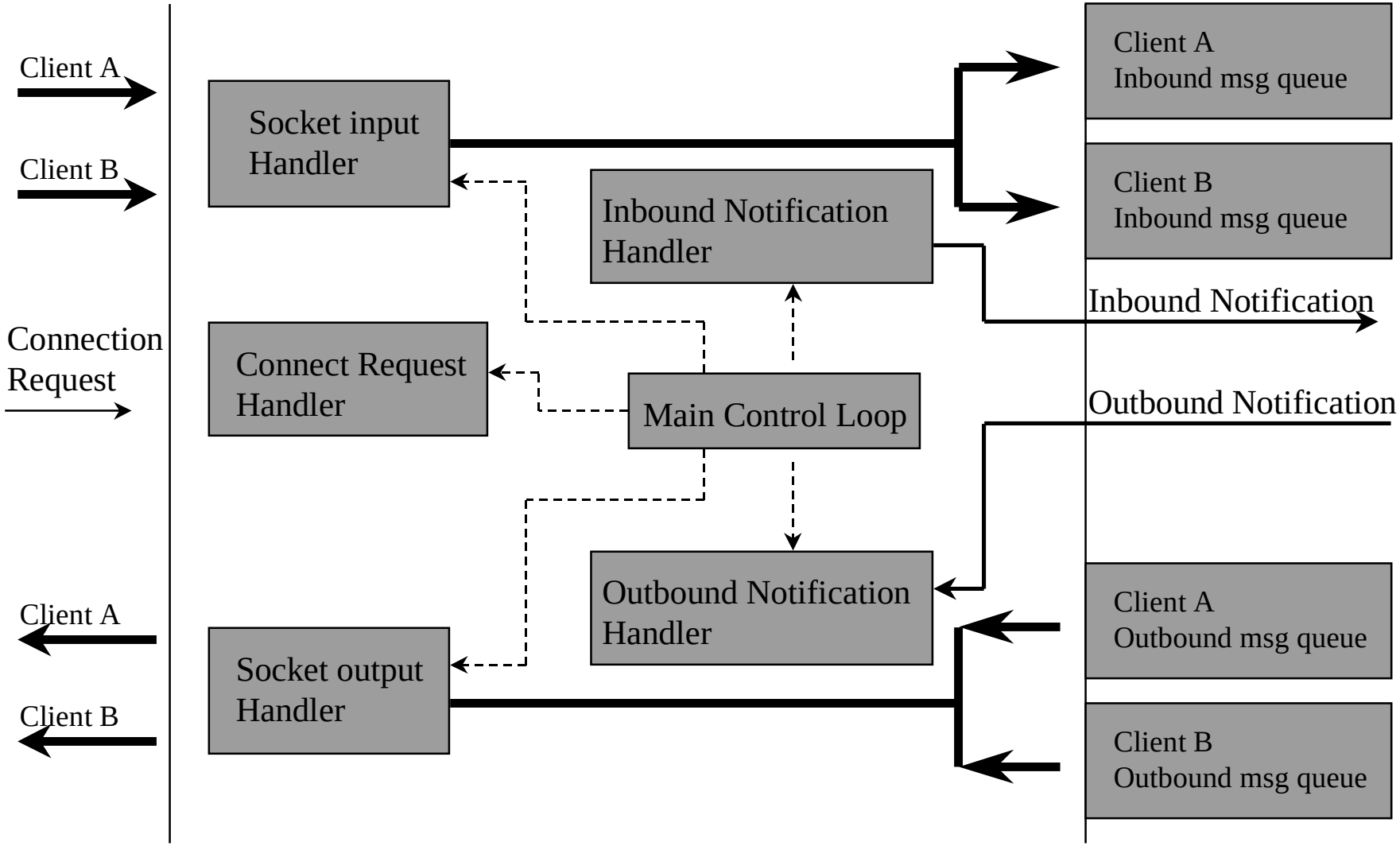
Interface Driver CSC

- Inputs
  - Raw services request messages from client processes
  - Services response messages from the Data Server proper
  - Command line arguments
  - Environment variables
- Outputs
  - Raw service response messages to client processes

Interface Driver CSC (cont)

- Startup
  - Initialize data structures
  - Open TCP listening port
  - wait for client connections
- Processing
  - Accept client connection when available
  - Read service request message from client
  - Notify Data Server of pending message
  - Read service response response message from Data Server
  - Forward message to appropriate client
- Shutdown
  - Close all client connections, close TCS listening port, release shared memory

Interface Driver CSC Block Diagram



DataServer API

- Description
  - Provides a small easy to use Application Programming Interface (API)
    - Calls for connecting and disconnecting to the DS
    - Create and destroy objects/buckets
    - Subscribe to objects
    - Calls for the accessing of object/bucket data fields via RADE
  - Interfaces
  - Inputs
    - Commands
    - Data

Representation and Description Engine (RADE) CSC

- Description
  - An application programming interface that encapsulates object descriptions allowing runtime changes to data objects
    - Provides item level access with type and range checking
    - Provides processor independent transmission format
    - Isolates the application from the object's underlying storage format
    - Objects may be changed without the need to rebuild an application